Jarkko Hammar

Verkkoliikenneanalyysi Pythonin tekoälykirjastojen avulla

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintätekniikka

Insinöörityö

7.4.2025

Tiivistelmä

Tekijä:	Jarkko Hammar
Otsikko:	Verkkoliikenneanalyysi Pythonin tekoälykirjastojen avulla
Sivumäärä:	22 sivua
Aika:	7.4.2025
Tutkinto:	Insinööri (AMK)
Tutkinto-ohjelma:	Tieto- ja viestintätekniikka
Ammatillinen pääaine:	Tietotekniikka
Ohjaajat:	Yliopettaja Janne Salonen

Tässä työssä tutkittiin koneoppimismallien suorituskykyä verkkohyökkäysten tunnistamisessa käyttämällä CIC-IDS-2017-datasettiä, joka sisältää sekä normaalia verkkoliikennettä että useita eri hyökkäystyyppejä, kuten DDoS, PortScan ja Brute Force. Tavoitteena oli vertailla kolmea erilaista mallia: logistinen regressio, Random Forest (RF) ja monikerroksinen neuroverkko (MLP). Vertailukriteereinä käytettiin tarkkuutta, F1-scorea ja mallien koulutusaikaa.

Kaikki mallit tunnistivat yleiset hyökkäykset (esim. DDoS, DoS Hulk, PortScan) lähes täydellisesti, mutta harvinaisten hyökkäysten (esim. XSS, SQL Injection, Brute Force) tunnistuksessa oli ongelmia erityisesti logistisella regressiolla. Random Forest suoriutui parhaiten myös harvinaisten hyökkäysten osalta. Tulokset osoittivat, että Random Forest on paras valinta verkkohyökkäysten tunnistamiseen, sillä se yhdistää korkean tarkkuuden ja kohtuullisen koulutusajan. Logistinen regressio oli nopein koulututtaa, mutta ei yhtä luotettava hyökkäysten tunnistamisessa. MLP tarjosi hyvän tunnistus tarkkuuden, mutta oli hidas kouluttaa.

Tämä työ osoittaa, että koneoppiminen voi olla tehokas työkalu verkkohyökkäysten torjunnassa, mutta parannuksia tarvitaan erityisesti harvinaisten ja monimutkaisten hyökkäysten tunnistuksessa.

Avainsanat: Python, verkkoliikenne, verkkoliikenneanalyysi, tekoäly, tekoälykirjasto, koneoppiminen

---

Abstract

Author:	Jarkko Hammar
Title:	Network Traffic Analysis Using Python's AI Libraries
Number of Pages:	22 pages
Date:	7 April 2025
Degree:	Bachelor of Engineering
Degree Programme:	Information and Communications Technology
Professional Major:	Information Technology
Supervisors:	Principal Lecturer Janne Salonen

This study investigated the performance of machine learning models in detecting network attacks using the CIC-IDS-2017 dataset, which contains both normal network traffic and various types of attacks, such as DDoS, PortScan, and Brute Force. The goal was to compare three different models: logistic regression, Random Forest (RF), and multilayer perceptron (MLP). The comparison criteria included accuracy, F1-score, and model training time.

All models identified common attacks (e.g., DDoS, DoS Hulk, PortScan) with near-perfect accuracy. However, there were challenges in detecting rare attacks (e.g., XSS, SQL Injection, Brute Force), especially with logistic regression. Random Forest performed the best, even for rare attacks. The results indicate that Random Forest is the optimal choice for detecting cyberattacks, as it combines high accuracy with a reasonable training time. Logistic regression was the fastest to train but less reliable in attack detection. MLP provided good detection accuracy but was slow to train.

This work demonstrates that machine learning can be an effective tool for combating network attacks, but improvements are needed, especially in detecting rare and complex attacks.

Keywords: Python, network traffic, networktraffic analysis, artificial intelligence, AI library, machine learning

Sisällys

Lyhenteet

Botnet: Verkko, joka koostuu useista haittaohjelmilla saastutetuista tietokoneista, joita hyökkääjä ohjaa keskitetysti tai hajautetusti. Bottiverkkoja käytetään usein haitallisiin toimintoihin kuten DDoS.

CSV: Tiedostomuoto, jossa arvot erotetaan pilkulla (Comma-Separated Values)

DoS: Palvelunestohyökkäys (engl. Denial of Service) voi kohdistua sivustoon, palvelimeen tai verkkoon, ja sen tarkoitus on häiritä sen toimintaa tai estää se kokonaan.

DDoS: Hajautettu palvelunestohyökkäys (engl. Distributed Denial of Service).

IDS: Tunkeutumisen havaitsemisjärjestelmä (engl. Intrusion Detection System). IDS-järjestelmät ovat keskittyneet seuraamaan verkkoliikennettä ja havaitsemaan mahdollisia tietoturvauhkia.

IPS: Tunkeutumisen estämisjärjestelmä (Intrusion Prevention System).

PCAP: PCAP – Pakettien tallennusmuoto (Packet Capture)

XSS: Cross-Site Scripting, tietoturvahaavoittuvuus, jossa haitallinen koodi lisätään verkkosivustolle ja suoritetaan käyttäjän selaimessa.

1 Johdanto

Internet ja tekoäly ovat suurimpia aikamme megatrendeistä ja niiden suosion mukana myös käyttäjämäärät ja vaatimukset tietoverkoille ovat nousseet ennennäkemättömälle tasolle. Digitaalisen kehityksen myötä kyberhyökkäyksistä on tullut merkittävä uhka ja tietoverkkojen turvallisuus on tärkeämpää kuin koskaan.

Tässä opinnäytetyössä tutkittiin, kuinka koneoppimista voidaan hyödyntää verkkohyökkäysten tunnistamisessa. Työssä käydään läpi verkkoliikennettä ja erilaisia verkkohyökkäyksiä yleisesti sekä keskitytään tekoälyyn ja tekoälytyökalujen yleistymiseen verkonhallinnassa. Siinä syvennytään myös IDS-järjestelmiin ja niiden ominaisuuksiin sekä haasteisiin.

Opinnäytetyössä hyödynnettiin Python-ohjelmointikieltä ja sen kirjastoja verkkoliikenteen analysointiin sekä hyökkäysdatan keräämiseen. Näiden tietojen pohjalta kehitettiin ja koulutettiin koneoppimismalli, joka erottaa normaalin verkkoliikenteen haitallisista yhteyksistä. Mallin suorituskykyä arvioitiin yleisesti käytetyillä mittareilla, kuten accuracy, precision ja F1-score, ja tulokset visualisoitiin Pythonin työkaluilla havainnollisuuden parantamiseksi. Lopuksi tehtiin yhteenveto tuloksista ja kokeen onnistumisesta.

Työn tavoitteena oli kehittää järjestelmä, joka pystyy tunnistamaan tunnettuja ja tuntemattomia verkkohyökkäyksiä. Lisäksi työssä käsiteltiin koneoppimismallien soveltamiseen liittyviä haasteita, kuten datan esikäsittelyä, ominaisuuksien valintaa ja mallin optimointia.

Insinöörityö tarjoaa kattavan katsauksen koneoppimisen hyödyntämiseen verkkoturvallisuudessa ja antaa pohjan jatkokehitykselle, jossa tekoälyä voidaan soveltaa entistä monipuolisemmin verkkohyökkäysten torjunnassa. Aihe valikoitui kiinnostuksestani tekoälyä ja sen kasvavia mahdollisuuksia kohtaan.

2 Verkkoliikenne

Joka päivä verkkoon lisätään noin kuusi miljoonaa uutta laitetta ja niistä jokainen on potentiaalinen tietoturvariski. (State of IoT 2024.) Ongelmia aiheuttavat heikot salasanat sekä vanhentuneet laitteet ja ohjelmistot. Tietoja voidaan urkkia tai haittaohjelma voi kaapata laitteet osaksi kyberhyökkäystä. Osa nykyisistä tietoverkoista perustuu vanhentuneisiin tekniikoihin, jotka aiheuttavat tietoturvariskejä, sillä ne eivät ole suunniteltu kestämään nykyistä laitemäärää ja yhteyskuormitusta. Lisäksi vanhojen verkkojen ylläpitokustannukset ovat kasvaneet merkittävästi, koska laitteet ja ohjelmistot vaativat jatkuvia korjauksia ja päivityksiä. (Tecnobits 2025.)

2.1 Verkon analysointi

Verkonanalysoinnilla tarkoitetaan verkkoliikenteen tarkkailua, tutkimista ja mittaamista. Sen avulla voidaan ymmärtää verkon toimintaa, suorituskykyä ja mahdollisia tietoturvauhkia sekä suojautua uhkilta ja kyberhyökkäyksiltä. Se on keskeinen osa verkonhallintaa ja verkkoturvallisuutta ja sitä käytetään sekä organisaatioiden sisäverkkojen että laajempien tietoverkkojen, kuten internetin, toiminnan valvontaan ja optimointiin (Rapid7 2025).

Verkon analysoinnilla voidaan saavuttaa monia tärkeitä tavoitteita. Verkon suorituskykyä voidaan optimoida, kun tunnistetaan viiveet, ruuhkat sekä pullonkaulat. Ongelmien ja vikojen selvitys helpottuu, kun nähdään mikä aiheuttaa hitautta tai yhteyskatkoksia ja samalla valvotaan laitteita kuten reitittimiä, kytkimiä ja palvelimia. Näiden lisäksi verkkohyökkäysten tunnistus nopeammin on mahdollista, kun havaitaan epätavallinen liikenne kuten hyökkäykset ja muut tunkeutumisyritykset. (Rapid7 2025.)

3 Verkkohyökkäykset

Verkkohyökkäykset ovat tietoverkkoihin tai tietojärjestelmiin kohdistuvia hyökkäyksiä, joiden tarkoituksena voi olla tietojen varastaminen, palveluiden häiritseminen, järjestelmien vahingoittaminen tai muu haitallinen toiminta. Niitä voivat suorittaa yksittäiset hakkerit, rikollisryhmät tai jopa valtiolliset toimijat. (Cisco 2025.) Yleisimpiä hyökkäystyyppejä ovat:

Denial of Service, DoS eli palvelunestohyökkäys. Verkkoa vastaan voidaan hyökätä kohdistamalla siihen massiivinen määrä liikennettä, mikä johtaa verkon ruuhkautumiseen ja käytännössä sen toiminnan lamaantumiseen. (Cisco 2025.)

Distributed Denial of Service, DDoS eli hajautettu palvelunestohyökkäys. Ruuhkauttavan liikenteen luomiseen käytetään usein myös monien kaapattujen laitteiden muodostamia bottiverkkoja (Botnet). (Cisco 2025.)

Brute Force- hyökkäyksessä tavoitteena on murtaa käyttäjänimi- ja salasanayhdistelmät kokeilemalla suuria määriä mahdollisia salasanoja automaattisesti. (Fortinet 2025.)

PortScan- hyökkäyksessä hyökkääjä skannaa avoimia portteja selvittääkseen, mitkä palvelut ovat käytössä ja voivatko ne olla haavoittuvia. (Fortinet 2025.)

SQL-injektio tapahtuu, kun hyökkääjä syöttää haitallista SQL-koodia verkkosivuston lomakkeisiin saadakseen pääsyn tietokantaan ja varastaakseen dataa. (Cisco 2025.)

4 Verkkoliikenteen analysoinnin työkalut

Verkkoliikenteen analysointiin on olemassa useita valmiita ohjelmia ja työkaluja. Niitä ovat esimerkiksi Wireshark, Nmap, Tcpdump ja Scapy (Kuva 1).

Suosituin näistä on Wireshark ja sitä käyttävät jopa IT-ammattilaiset ja järjestelmänvalvojat. Se on paras vaihtoehto verkkoliikenteen graafiseen analysointiin ja visuaaliseen tarkasteluun, mutta sillä ei voi lähettää paketteja. (Wireshark 2025.)

Nmap sopii erityisesti verkon skannaukseen ja haavoittuvuuksien kartoittamiseen, mutta sillä ei voi kaapata paketteja. (Nmap 2025.)

Tcpdump on kevyt ja nopea komentorivityökalu verkkoliikenteen kaappaamiseen, mutta siinä ei ole graafista käyttöliittymää ja se vaatii käyttäjältä komentoriviosaamista. (Tcpdump 2025.)

Scapy on Python-kirjasto verkkoliikenteen analysointiin ja eroaa muista erityisesti siinä, että se ei ole pelkästään analysointityökalu. Sillä pystyy myös luomaan ja muokkaamaan paketteja ja simuloimaan verkkohyökkäyksiä mikä tekee siitä erittäin hyödyllisen tietoturva- ja verkkotestauksessa. Scapy on joustavampi ja tehokkaampi kuin monet muut verkkoanalysointiohjelmat, mutta sen käyttö edellyttää ohjelmointitaitoja. (Scapy 2025.)

Ominaisuus	Scapy	Wireshark	Nmap	Tcpdump
Käyttötarkoitus	Pakettien luonti ja analyysi	Verkkoliikenteen analyysi	Verkkoskannaus	Verkkoliikenteen kaappaus
Ohjelmointimahdollisuus	Kyllä (Python)	Ei	Ei	Ei
Graafinen käyttöliittymä	Ei	Kyllä	Ei	Ei
Pakettien muokkaus ja lähetys	Kyllä	Ei	Ei	Osittain
Reaaliaikainen analyysi	Kyllä	Kyllä	Osittain	Kyllä

Kuva 1. Verkkotyökalujen ja niiden ominaisuuksien vertailu taulukossa

Wireshark, Tcpdump ja Scapy tallentavat verkkoliikenteen PCAP-tiedostoina, mikä on pääformaatti verkkoliikenteen kaappauksille. Tiedostot voivat olla hyvin suuria ja koneoppimismalleja varten ne käsitellään koneoppimismalleille sopivimmiksi CSV-tiedostoiksi (Wireshark 2025). Nmap tallentaa tiedostot yleensä suoraan normaalina tekstitiedostona (.txt). (Nmap 2025.)

Verkkoliikenteen analysointi voidaan tehdä manuaalisesti tai automaattisesti tekoälypohjaisilla järjestelmillä. Verkon analysointi aloitetaan tavoitteiden määrittelyllä. Niitä voivat olla verkon suorituskyvyn optimointi, epänormaalin liikenteen, kuten hyökkäysten, tunnistaminen tai verkkoliikenteen vianmääritys. Kun tavoitteet on määritelty valitaan verkkoliitäntä, josta liikenne kaapataan. Tähän käytetään jotain edellä mainituista paketinkaappausohjelmista. Lopuksi suodatetaan paketit, joita halutaan tarkastella esim. IP-osoitteet tai portit. (Schule 2025.)

Tätä seuraa itse datan analysointi, josta nähdään normaali liikenne ja epänormaali liikenne, kuten suuret tiedonsiirtopiikit, tuntemattomat IP-osoitteet ja epäilyttävät portit. Datasta nähdään myös mahdollinen haitallinen liikenne kuten DoS-hyökkäykset, skannaukset ja haittaohjelmat. Analyysissä voidaan hyödyntää ohjelmien työkaluja ja suodattimia, jotka ovat käteviä nopeuttamaan analyysiä. Myös automaattinen analysointi IDS-järjestelmien ja tekoälyn avulla on mahdollista. Kun analyysi on tehty, voidaan estää epäilyttävä liikenne (esim. palomuurisäännöillä), optimoida verkon suorituskykyä ja luoda raportteja sekä dokumentoida löydökset. (Schule 2025.)

Verkkoliikennedataa on yleisesti vaikeaa analysoida sillä se sisältää monimutkaisia riippuvuuksia ja epälineaarisia ilmiöitä. Verkkohyökkäykset voivat olla satunnaisia tai perustua tiettyihin kuvioihin. Osa verkkodatan muuttujista, kuten pakettien määrä, latenssi tai yhteyksien määrä, ei noudata lineaarisia malleja. Joissakin tilanteissa pieni muutos yhdessä muuttujassa voi aiheuttaa suuren vaikutuksen toisaalla. Lisäksi erilaiset protokollat, portit ja liikennemäärät voivat vaikuttaa toisiinsa monimutkaisilla tavoilla. Esimerkiksi DDoS-hyökkäykset voivat aiheuttaa eksponentiaalisen kasvun tietyissä muuttujissa ja myös

normaalissa verkkoliikenteessä esiintyy piikkejä ja jaksottaisia trendejä. (Khraisat ym. 2019.)

5 Tekoäly

Tekoälyllä tarkoitetaan koneen kykyä suorittaa ihmiselle tyypillisiä taitoja kuten ongelmien ratkaisua ja uuden oppimista. Koneoppiminen on järjestelmä, joka hyödyntää dataa oppimiseen, päätöksentekoon ja toimintansa kehittämiseen. Se perustuu algoritmeihin, jotka pystyvät toimimaan itsenäisesti ja mukauttamaan käyttäytymistään. Koneoppiminen jaetaan valvottuun oppimiseen, valvomattomaan oppimiseen ja vahvistusoppimiseen. Syväoppiminen on koneoppimisen alalaji, joka pystyy ratkaisemaan monimutkaisia ja päättelyä vaativia tehtäviä. Sen kyky käsitellä luonnollista dataa, kuten tekstiä, kuvia ja videoita, mahdollistaa esimerkiksi tekstin kääntämisen ja kuvien luokittelun. (Marchal, Nawrotech, WithSecure 2024.)

5.1 Tekoäly tietoturvassa

Perinteiset kyberpuolustuksen menetelmät analysoivat suuria datamääriä hitaasti ja tuottavat runsaasti vääriä hälytyksiä, mikä kuormittaa datan käsittelyä. Nopeasti kehittyvä tekoäly on yhä keskeisempi osa kyberturvallisuutta. Tekoälyä hyödynnetään erityisesti tehtävissä, jotka vaativat nopeaa ennustamista, tiedonkäsittelyä ja poikkeamien havaitsemista. Esimerkiksi tietoverkoissa tapahtuvan epäilyttävän toiminnan tunnistaminen ja siihen reagointi tehostuvat tekoälyn avulla, mikä mahdollistaa kyberuhkien nopean havaitsemisen, torjumisen ja ennaltaehkäisyn. (Vähäkainu, Lehto & Neittaanmäki 2018.)

Toisaalta tekoälyä hyödyntämällä voidaan luoda todella nopeasti erilaisia verkkohyökkäyksiä ja luotettavan näköisiä huijausviestejä. Tekoäly osaa kehittää kyberhyökkäyksiä yhä monimutkaisemmiksi ja tehokkaammiksi. (Keeper 2024.) Erilaiset verkkohyökkäykset, kuten palvelunestohyökkäykset (Dos),

haittaohjelmien levitys ja tietomurrot aiheuttavat merkittäviä taloudellisia ja toiminnallisia haittoja yksityishenkilöille ja organisaatioille. Esimerkiksi Britanniassa kyberhyökkäykset ovat aiheuttaneet yrityksille noin 44 miljardin punnan (55,08 miljardin dollarin) tulonmenetykset viimeisen viiden vuoden aikana ja 52 % yksityisen sektorin yrityksistä on raportoinut vähintään yhden hyökkäyksen tänä aikana. (Reuters 2024.)

Perinteiset tietoturvamenetelmät, kuten palomuurit ja tunnistussäännöillä toimivat järjestelmät eivät enää riitä torjumaan uusia ja kehittyneitä hyökkäysmuotoja. Hyökkäysten edistyksellisyys näkyy myös esimerkiksi niiden kyvyssä vältellä havaitsemista sekä tunnistaa haavoittuvuuksia suojausjärjestelmissä ja tietoverkoissa. Tämän vuoksi koneoppimiseen perustuvien järjestelmien käyttö verkkohyökkäysten tunnistamisessa on noussut keskeiseksi tutkimusalueeksi ja kehityskohteeksi. (Khraisat ym. 2019.)

6 Tunkeutumisen havaitsemisjärjestelmä IDS

Yksi merkittävimmistä edistysaskeleista tekoälyn hyödyntämisessä kyberpuolustuksessa on tunkeutumisen havaitsemisjärjestelmien (Intrusion Detection Systems, IDS) kehittyminen. IDS-järjestelmät ovat keskittyneet seuraamaan verkkoliikennettä ja havaitsemaan mahdollisia tietoturvauhkia. Järjestelmä ilmoittaa kaikista tunkeutumistoimista tai rikkomuksista järjestelmänvalvojalle. (Khraisat ym. 2019.)

IDS-järjestelmät voivat olla allekirjoituspohjaisia tai anomaliapohjaisia. Allekirjoituspohjaiset perustuvat tunnetuille hyökkäysmalleille ja säännöille ja niillä voi olla tunnettujen haittaohjelmien kirjastoja järjestelmässään. Uusien hyökkäyksien sattuessa tietokantaan lisätään uusi allekirjoitettu datamalli perustuen tapahtuneeseen hyökkäykseen. Anomaliapohjaiset oppivat normaalin verkkoliikenteen aiemmin luotujen mallien perusteella ja havaitsevat poikkeamat sekä mahdolliset hyökkäykset. Ne perustuvat usein koneoppimiseen. (Khraisat ym. 2019.)

Molempien järjestelmien toiminta perustuu poikkeamien havaitsemiseen tietoverkkoliikenteessä. Järjestelmät voivat käyttää erilaisia menetelmiä liikenteen kartoittamiseen kuten pakettianalyysi ja liikenneanalyysi. Pakettianalyysilla tarkoitetaan yksittäisten verkkopakettien kaappaamista ja tutkimista. Haitalliset tai epätavalliset paketit tunnistetaan ja niistä annetaan hälytys. Verkkoliikenne analyysissä tarkastellaan kokonaisliikenteen määrää ja havaitaan haitallisia liikennemalleja kuten DDoS- hyökkäyksiä tai liikennepoikkeamia, kuten äkilliset tiedonsiirtopiikit tai epänormaalit yhteydet. (Khraisat ym. 2019.)

Peruslähestymistapa on luoda koneoppimisen avulla malli luotettavasta toiminnasta ja sitten vertailla normaalista poikkeavaa käyttäytymistä tähän malliin. Koneoppimispohjaisen analyysin yhdistäminen IDS- järjestelmään auttaa parantamaan tunnistustarkkuutta ja uusia uhkia voidaan havaita tehokkaammin. Tekoälyyn pohjautuvat IDS-järjestelmät hyödyntävät muun muassa poikkeamien tunnistamista sekä koneoppimisen valvottua ja valvomatonta oppimista. Tämä teknologia perustuu tekoälyalgoritmeihin, jotka parantavat järjestelmän tehokkuutta ja tarkkuutta, vähentäen samalla virheellisiä havaintoja. Sen avulla voidaan analysoida suuria tietomääriä ja havaita monimutkaisia hyökkäysmalleja, joita perinteinen IDS ei tunnista. Tekoälyperusteisten ratkaisujen kehittyminen IDS-järjestelmissä on helpottanut uusien, innovatiivisten ja muuttuvien uhkien havaitsemista kuten nollapäivähyökkäysten, jotka hyödyntävät ennestään tuntemattomia järjestelmähaavoittuvuuksia. (Khraisat ym. 2019.)

6.1 IDS-järjestelmien haasteet

Tekoälypohjaisten IDS-järjestelmien käyttöön liittyy kuitenkin haasteita, kuten koneoppimismallien opettaminen liian yksipuolisella datajoukolla, mikä voi johtaa vääristyneisiin havaintoihin. Haavoittuvuuksien hallinnan tavoitteena on tunnistaa järjestelmien tietoturvan heikkoudet ennen mahdollisten kyberuhkien toteutumista. Tekoälyä hyödyntävät IDS-järjestelmät automatisoivat vikojen ja turvallisuusaukkojen havaitsemisen koneoppimisalgoritmien avulla, jotka mahdollistavat järjestelmien jatkuvan valvonnan sekä nopeat toimenpiteet havaittujen puutteiden ilmetessä. (Khraisat ym. 2019.)

Kuitenkin vaikka allekirjoituspohjainen järjestelmä tunnistaa helposti tunnetut hyökkäykset se ei välttämättä tunnista uusia tai muunneltuja hyökkäyksiä. Anomaliapohjainen järjestelmä taas täytyy kouluttaa erikseen suurella määrällä dataa mikä vaatii resursseja ja aikaa sekä työvoimaa. IDS-järjestelmät eivät välttämättä anna tarpeellista kontekstia mahdollisesta hyökkäyksestä, ja järjestelmänvalvoja joutuu tekemään suuren määrän työtä selvittäessään annettua hälytystä. Usein hälytykset voivat olla myös täysin vääriä mikä on yksi suurimmista ongelmista. IDS-järjestelmät eivät myöskään pysty pysäyttämään havaitsemiaan hyökkäyksiä vaan se jää työntekijöiden harteille (Intrusion Detection System 2025). Jotkut IDS-tuotteet pystyvät reagoimaan havaittuihin tunkeutumisiin ja pysäyttämään ne itsenäisesti. Reagointikykyisiä järjestelmiä kutsutaan tyypillisesti tunkeutumisen estojärjestelmäksi (IPS). (Khraisat ym. 2019.)

7 Python ja tekoälykirjastot

7.1 Python

Python on korkean tason tulkattava ja monipuolinen ohjelmointikieli. Sitä käytetään monilla eri aloilla kuten datatieteessä, tietoturvassa, tekoälyssä, automaatiossa ja verkkoliikenne analyysissä. Se on yksi suosituimmista ohjelmointikielistä tekoäly- ja koneoppimisprojekteissa ja sen suosio perustuu selkeään syntaksiin, laajaan yhteisötukeen sekä monipuolisiin kirjastoihin. Kirjastot tarjoavat tehokkaita työkaluja datan mallintamiseen, analysointiin ja neuroverkkojen rakentamiseen. Lisäksi python tukee sekä perinteisiä koneoppimismenetelmiä että syväoppimista, mikä tekee siitä monipuolisen työkalun tekoälyn eri osa-alueilla. Pythonin avoimen lähdekoodin ekosysteemi mahdollistaa nopean kehityksen ja kokeilun eri tekoälyprojekteissa. Se tarjoaa pääsyn moniin valmiisiin algoritmeihin, visualisointityökaluihin ja laskentakirjastohin, jotka helpottavat monimutkaisten mallien kehittämistä ja optimointia. (Python 2025.)

7.2 Tekoälykirjastot

Python tarjoaa laajan kokoelman kirjastoja, jotka helpottavat koneoppimisen ja tekoälyn kehittämistä. Lisäksi Pythonilla on vahva ja aktiivinen yhteisö, joka tarjoaa runsaasti valmiita ratkaisuja, opetusmateriaaleja sekä dokumentaatiota.

Suosituimpia tekoälykirjastoja Pythonissa ovat:

Numpy: Tehokas kirjasto numeeriseen laskentaan, eritysesti matriisioperaatioihin ja lineaaroalgebraan. (Best Python libraries for Machine Learning 2024.)

Pandas: Datan esikäsittelyn ja analysointiin erikoistunut kirjasto, joka tarjoaa tehokkaat työkalut taulukkomuotoisen datan käsittelyyn, erityisesti datan läpikäynti, muokkaaminen ja tilastollinen analyysi (Best Python libraries for Machine. Learning 2024.)

Matplotlib: Visualisointikirjasto, jonka avulla voidaan luoda monipuolisia kaavioita ja graafeja. Käytetään usein Pandasin tai Numpyn kanssa datan tutkimiseen. (Best Python libraries for Machine Learning 2024.)

Scikit-learn: Koneoppimisen perustyökalu, joka sisältää laajan valikoiman valmiita algoritmeja luokitteluun, regressioon sekä klusterointiin. Sisältää myös tärkeitä esikäsittely- ja mallinoptimointityökaluja, kuten ominaisuuksien normalisointia ja hyperparametrien viritystä. Tukee erilaisia oppimismenetelmiä, kuten satunnaismetsiä, tukivektorikoneita (SVM) ja päätöspuita. (Best Python libraries for Machine Learning 2024.)

Tensorflow: Googlen kehittämä kirjasto syväoppimiseen ja neuroverkkojen rakentamiseen. (Best Python libraries for Machine Learning 2024.)

PyTorch: Metan kehittämä kirjasto, joka tarjoaa joustavamman lähestymistavan neuroverkkojen kehittämiseen. (Best Python libraries for Machine Learning 2024.)

OpenCV: Tietokonenäköä varten kehitetty kirjasto, joka tarjoaa työkalut kuvien ja videoiden analysointiin. Käytetään kasvojentunnistuksessa, liikenteen seurannassa ja lääketieteellisessä kuvantamisessa. (OpenCV 2025.)

Pythonin tekoälykirjastot tarjoavat monia etuja ja ratkaisuja, mutta niihin liittyy myös haasteita. Ne vaativat suuren määrän laskentatehoa ja niiden luomat mallit voivat olla monimutkaisia ymmärtää.

8 Ympäristön valmistelu

Työ suoritettiin käyttämällä Pythonia, sen kirjastoja ja tekoälyratkaisuja. Työn tavoitteena oli luoda automaattinen järjestelmä, joka tunnistaa epänormaalia verkkoliikennettä ja mahdollisia tunkeutumisyrityksiä. Toteutuksessa hyödynnettiin Pythonin tekoälykirjastoja.

Ensin asennettiin tarvittavat ohjelmistot ja kirjastot, jotka mahdollistavat verkkoliikenteen kaappaamisen, datan analysoinnin sekä tekoälykirjastojen hyödyntämisen. Asennettavia ohjelmistoja olivat: Python 3.13.2, Numpy ja Pandas datan käsittelyyn, Scikit-learn koneoppimismallien kehittämiseen, Matplotlib ja Seaborn tulosten visualisointiin (Esimerkkikoodi 1).

```
pip install numpy pandas scikit-learn matplotlib seaborn
```

Esimerkkikoodi 1. Pythonin komentorivillä suoritettava käsky, joka asentaa useita koneoppimisee ja data-analyysiin liittyviä kirjastoja pip-paketinhallinnan avulla.

8.1 Käytettävä datasetti

Työssä käytettiin valmista datasettiä CIC-IDS2017 (Intrusion detection evaluation dataset (CIC-IDS2017)). Datasetti on suunniteltu erityisesti verkkohyökkäysten tunnistamiseen, ja se sisältää laajan valikoiman erilaisia tunkeutumisyrityksiä. Datasetti on jaettu eri tiedostoihin, jotka sisältävät normaalia verkkoliikennettä sekä erillisiä hyökkäysten esimerkkejä. Se sopii erinomaisesti koneoppimismallien ja neuroverkkojen koulutukseen ja arviointiin. Se kattaa myös

liikenteen aikatiedot ja protokollatiedot, jotka voivat olla hyödyllisiä liikenteen analysoimisessa ja poikkeamien tunnistamisessa. Datasetti valittiin useista syistä. Ensinnäkin se tarjosi laajan ja monipuolisen aineiston, joka sisälsi useita eri verkkohyökkäyksiä, kuten DDoS- ja SQL-injektioita, sekä normaalia verkkoliikennettä. Tämä mahdollisti tehokkaan ja kattavan mallin kouluttamisen. Käyttämällä valmista datasettiä pystyttiin keskittymään analyysiin ja mallin kehittämiseen. Datasetin on julkaissut Canadian Institute for Cybersecurity (CIC) helmikuussa 2017. Tiedot on esikäsitelty valmiiksi PCAP-muodosta CSV-muotoon, joka on optimoitu koneoppimismallien koulutukseen.

8.2 Datasetin valmistelu

Kahdeksan erilaista verkkoliikennettä sisältävää CSV-tiedostoa yhdistettiin yhdeksi combined-datasetiksi, jota käytettiin aineistona koneoppimismalleille. Osa niistä sisälsi normaalia liikennettä ja osa hyökkäyksiä kuten DDoS, PortScan ja Brute Force.

Sen jälkeen määriteltiin tavoite, eli mitä malli yrittää ennustaa - onko kyseessä normaali liikenne vai hyökkäys. Tämä tieto löytyy usein valmiina datasetistä, kuten myös tässä tapauksessa.

9 Koneoppimismallien koulutus

Aluksi valittiin kolme koneoppimismallia, jotka koulutettiin ja testattiin verkkoliikenteessä esiintyvien hyökkäysten tunnistamiseksi. Käytettiin Scikit-learn-kirjaston luokittelumalleja: Logistinen regressio, Random Forest ja MLP-neuroverkko (multilayer perception, MLP).

Jokainen malli koulutettiin samalla esikäsitellyllä datalla. Tekoälymallin kouluttaminen tarkoittaa prosessia, jossa malli opetetaan tunnistamaan kaavoja ja tekemään ennusteita käyttämällä historiallista dataa. Tämä tapahtuu syöttämällä mallille opetusdataa, jonka perusteella se säätää painotuksiaan ja parametrejään optimoidakseen ennustustarkkuuden. Koulutusprosessin aikana mallia

arvioidaan erilaisilla mittareilla, kuten accuracy, precision ja F1-score, ja sitä hienosäädetään parempien tulosten saavuttamiseksi. Koodin rakenne oli samankaltainen kaikille malleille, mutta jokaisella mallilla oli omat eronsa koulutus- ja ennustusvaiheessa.

9.1 Koodin rakenne

Otettiin käyttöön Pandas, NumPy, Scikit-learn ja muut tarvittavat kirjastot (Esimerkkikoodi 2).

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, label_binarize
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report,
f1_score, confusion_matrix, roc_curve, auc
```

Esimerkkikoodi 2. Python-koodin alustusosa, jossa tuodaan tarvittavat kirjastot.

Datasetti ladattiin Pandas-kirjastolla minkä jälkeen poistettiin epäolennaiset sarakkeet. Puuttuvat arvot käsiteltiin (korvattiin keskiarvolla/mediaanilla) ja normalisoitiin data, jotta mallien oppiminen olisi tehokasta. Jaettiin 80/20-suhteessa koulutus- ja testijoukkoon. Ominaisuudet skaalattiin StandardScalerilla, jotta eri muuttujien arvot olisivat vertailukelpoisia (Esimerkkikoodi 3).# 1. Ladataan data

```
df = pd.read_csv(r'C:\Users\cleaned_combined_dataset_fixed.csv')

# 2. Datan jakaminen ominaisuuksiin (X) ja kohdemuuttujaan (y)
X = df.drop('Label', axis=1)
y = df['Label']

# 3. Datan jakaminen koulutus- ja testijoukkoihin
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42, stratify=y)

# 4. Ominaisuuksien skaalaaminen
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Esimerkkikoodi 3. Koodin osa, joka valmistelee datan koneoppimismallia varten.

Jokainen malli koulutettiin samalla esikäsitellyllä datalla. Logistinen regressio perustuu lineaariseen päätösrajapintaan. Random Forest käyttää useita päätöspuita ja yhdistää niiden tulokset. MLP-neuroverkko on syväoppimismalli, joka oppii monimutkaisempia rakenteita (Esimerkkikoodi 4).

```
# 5. Mallin valinta ja koulutus (Random Forest)
start_time = time.time()  # Aikaleima ennen koulutusta
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train_scaled, y_train)
```

Esimerkkikoodi 4. Koodin osa, joka luo ja kouluttaa Random Forest -mallin verkkoliikenteen hyökkäysten tunnistamiseen.

Jokainen malli testattiin koulutuksen jälkeen. Laskettiin tarkkuus, F1-score, precision ja recall. Tämän jälkeen ennusteet tallennettiin ja ana-lysoitiin. Tulokset tallennettiin tiedostoihin ja tehtiin confusion matrix -kuvat sekä heatmap- ja pylväsdiagrammivertailut (Esimerkkikoodi 5).# 6. Ennustaminen testijoukolla

```
y_pred = model.predict(X_test_scaled)
y_pred_proba = model.predict_proba(X_test_scaled)

# 7. Mallin arviointi
accuracy = accuracy_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred, average='weighted')
report = classification_report(y_test, y_pred)

# 8. Tallennetaan tulokset tiedostoon
with open('model_results_rf3.txt', 'w', encoding='utf-8') as f:
    f.write("Tarkkuus: {:.4f}\n".format(accuracy))
    f.write("F1-score: {:.4f}\n".format(f1))
    f.write("Koulutusaika: {:.2f} sekuntia\n".format(training_time))
    f.write("Luokitteluraportti:\n")
    f.write(report)

# 9. Tallennetaan ennusteet testijoukolle
output = pd.DataFrame({'True Labels': y_test, 'Predictions': y_pred})
output.to_csv('predictions_rf.csv', index=False)

# 10. Confusion Matrix
plt.figure(figsize=(8, 6))
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xtick-
labels=np.unique(y), yticklabels=np.unique(y))
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.title('Confusion Matrix - Random Forest')
plt.savefig('confusion_matrix_rf.png')
```

Esimerkkikoodi 5. Koodin osa, joka arvioita koulutetun Random Forest-mallin suorituskykyä ja tallentaa tulokset sekä visualisoi ne.

10 Mallien suorituskyky ja vertailu

10.1 Logistinen regressio

Ensin koulutettiin logistinen regressiomalli. Se on yksinkertainen, lineaarinen malli, joka toimii hyvin, kun eri luokkien välillä on selkeästi erottuvat rajat. Malli soveltuu erityisesti binääriseen luokitteluun ja perustuu todennäköisyyksiin.

Kouluttaminen oli nopeaa, mutta yllättäen se käytti paljon prosessoritehoa ja muistia, vaikka mallin pitäisi olla suhteellisen kevyt. Lineaarinen malli ei ole paras valinta monimutkaisiin, epälineaarisiin riippuvuuksiin, joita verkkoliikennedata yleisesti sisältää. Logistinen regressio ei myöskään suoriudu hyvin, jos eri hyökkäystyypit eivät ole helposti erotettavissa toisistaan. Dataa yritettiin tasapainottaa SMOTE:lla (Synthetic Minority Over-sampling Technique), mutta datasetti oli liian suuri, ja prosessi olisi kestänyt huomattavan kauan, joten ohjelma päätettiin keskeyttää.

10.2 Random Forest

Toisena koulutettiin Random Forest-malli. Malli luo useita päätöspuita, joista jokainen saa satunnaisen osan datasta. Satunnaisesti rakennetut päätöspuut yhdisttävät niiden tulokset (äänten enemmistö) lopulliseen päätökseen. Lopullinen ennuste saadaan äänestämällä tai keskiarvona. Random Forestilla on vahva kyky hallita epätasapainoista dataa, eli se ei ole yhtä herkkä sille, että jotkut luokat esiintyvät paljon useammin kuin toiset (esim. harvinaiset hyökkäykset). Se on yksi tämän hetken suosituimmista ja tehokkaimmista koneoppimismalleista.

Random Forest -mallin kouluttaminen oli hitaampaa kuin logistisen regression, mutta se vaati vähemmän resursseja ja suorituskykyä laitteistolta. Malli tuotti todella hyvät tulokset ja tunnisti lähes kaikki hyökkäykset. Ainoa hyökkäystyyppi, jonka kanssa mallilla oli pieniä ongelmia, olivat Brute Force -hyökkäykset.

10.3 MLP-neuroverkko

Viimeisenä koulutettiin MLP-neuroverkko. MLP on keinotekoinen neuroverkko, syväoppimismenetelmä, joka käyttää syötteitä, piilotettuja kerroksia ja ulostulokerrosta. Data kulkee kerrosten läpi, joissa jokainen neuroni tekee painotettuja laskuja ja käyttää aktivointifunktioita (kuten sigmoid). Sitä käytetään syvempien mallien yksinkertaisempana versiona. Toisin kuin Random Forest tai logistinen regressio, MLP voi oppia epälineaarisia rakenteita. Sen optimointi kuitenkin vaatii hyperparametrien kuten oppimisnopeuden ja kerrosten määrän säätöä. Lisäksi se on herkkä epätasapainoiselle datalle ja ylioppimiselle, jos sitä ei ole optimoitu.

MLP-neuroverkon koulutus kesti pidempään kuin muiden mallien koulutus. Toisaalta ja yllättävää kyllä, se käytti vähiten laitteiston resursseja ja oli kevyin kouluttaa. Pitkä koulutusaika voi johtua siitä, että malli ei jostain syystä pystynyt hyödyntämään laitteiston resursseja, vaikka niitä olisi ollut saatavilla. MLP-malli kuitenkin tuotti hyviä tuloksia ja tunnisti lähes kaikki hyökkäykset. Se ei kuitenkaan pärjännyt Random Forest -mallille, sillä mallilla oli suuria vaikeuksia tunnistaa Bot- ja Brute Force -hyökkäykset. Niitä se tunnisti erittäin huonosti ja luokitteli suurimman osan hyökkäyksistä turvalliseksi liikenteeksi.

11 Tulosten analysointi ja visualisointi

Tulokset analysoitiin Scikit-learn kirjaston työkaluilla ja visualisoitiin Matplotlib- ja Seaborn-kirjastoilla havainnollistamaan mallien havaintotarkkuutta. Visualisointi auttoi tunnistamaan liikenteessä olevat poikkeavuudet, jotka voivat viitata mahdollisiin hyökkäyksiin.

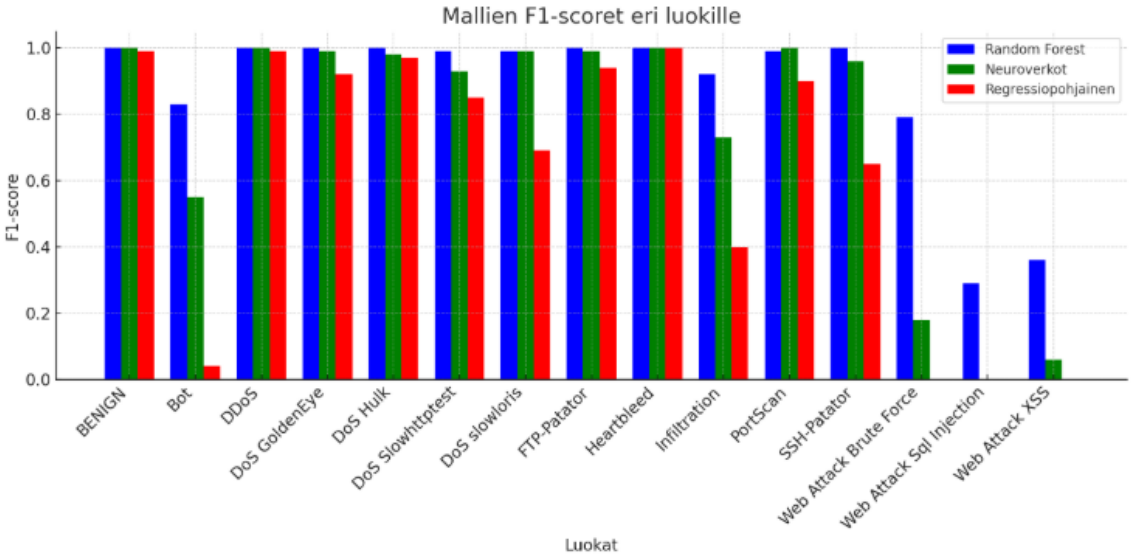
Mallien suorituskykyä arvioitiin precisionin, recallin, F1-scoren ja accuracy-mittarin avulla, jotka esitettiin luokitteluraportissa. Precision kertoo, kuinka usein malli oli oikeassa, kun se ennusti tietyn luokan (esimerkiksi hyökkäyksen). Toisin sanoen se mittaa, kuinka moni mallin tunnistamista hyökkäyksistä oli todellisia hyökkäyksiä. Recall kertoo, kuinka monta todellista hyökkäystä malli

onnistui tunnistamaan kaikista hyökkäyksistä. F1-score yhdistää precisionin ja recallin, ja se on erityisen hyödyllinen silloin, kun luokkajakauma on epätasainen (esim. hyökkäyksiä esiintyy harvoin). Accuracy kertoo, kuinka suuri osa mallin ennusteista oli oikein (positiiviset ja negatiiviset). Se voi kuitenkin olla harhaanjohtava, jos data on epätasapainossa ja tietty luokka esiintyy erittäin usein.

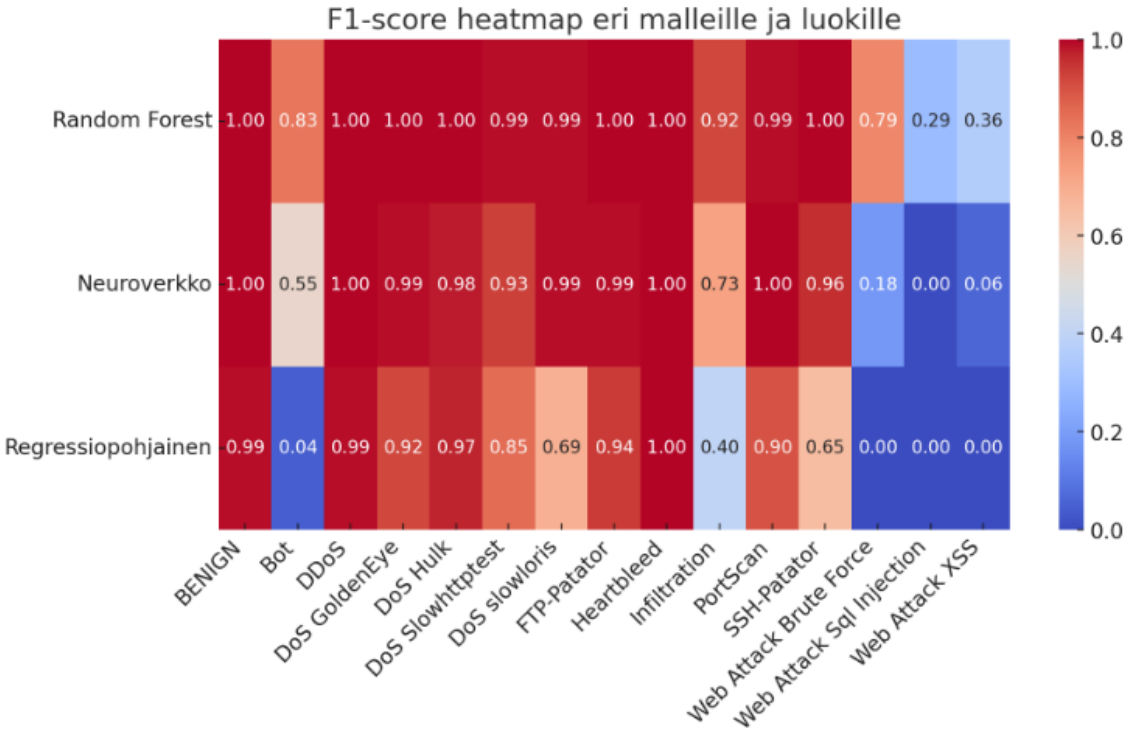
Yleisesti havaittiin, että Random Forest saavutti parhaan tarkkuuden, koska se hyötyi puiden yhdistelmästä ja pystyi käsittelemään monimutkaisia riippuvuuksia.

Logistinen regressio ei toiminut hyvin monimuotoisen datan kanssa, koska se ei pysty mallintamaan epälineaarisia suhteita.

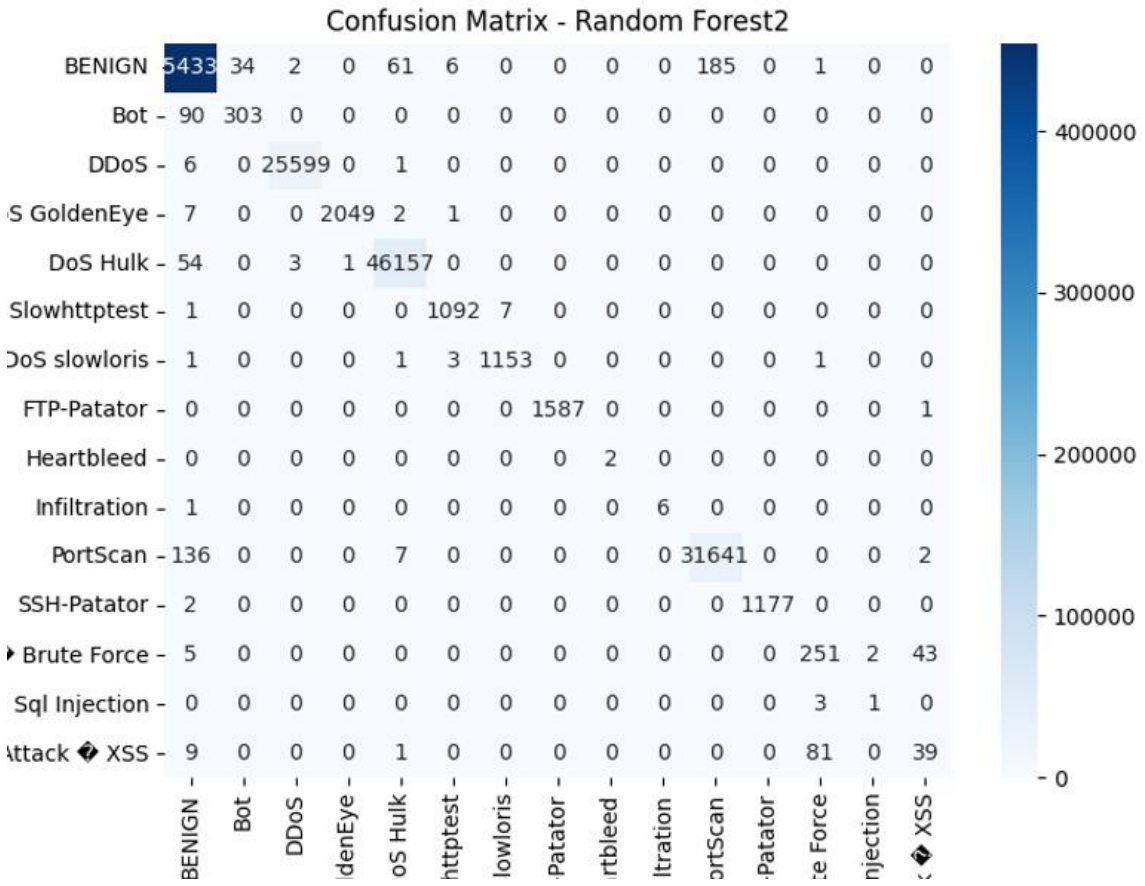
MLP-neuroverkko suoriutui kohtalaisesti, mutta se vaati paljon hienosäätöä ja oli hitaampi kouluttaa. Lisäksi mallin haasteina olivat ylikoulutus ja yleisimpien luokkien painotus. Ylikoulutuksella (overfitting) tarkoitetaan, että malli on oppinut liian hyvin koulutusdatan erityispiirteet, eikä se pysty yleistämään oppimaansa hyvin uusiin, tuntemattomiin datoihin. Yleisimpien luokkien painotuksella taas tarkoitetaan, että mallin koulutusdatassa on enemmän esimerkkejä tietyissä luokissa (esim. normaali liikenne) ja vähemmän harvinaisissa luokissa (esim. Brute Force -hyökkäykset). Tässä tapauksessa malli saattaa alkaa ennustamaan useammin yleisiä luokkia, jolloin se ei ole yhtä tarkka harvinaisempien luokkien tunnistamisessa. Näiden tekijöiden seurauksena mallin tunnistustarkkuus heikkenee, erityisesti harvinaisemmilla hyökkäystyypeillä.



Kuva 2.Eri koneoppimismallien F1-score-arvot eri hyökkäystyypeille. Kuvio luotu Pythonin Matplotlib- ja Seaborn-kirjastoilla.



Kuva 3. F1-score heatmap eri koneoppimismallien ja hyökkäystyyppien välillä. Punainen väri tarkoittaa korkeampaa F1-scorea, kun taas sininen ja vaaleat sävyt viittaavat huonompaan suoritukseen. Visualisointi toteutettu Pythonilla.



Kuva 4. Random Forest -mallin sekavuusmatriisi. Matriisi näyttää mallin ennusteiden tarkkuuden eri luokille: diagonaalilla ovat oikeat ennusteet, kun taas muut arvot edustavat virheellisiä luokitteluja. Kuvio luotu Pythonin Seaborn-kirjastolla.

11.1 Johtopäätökset

Random Forest oli paras valinta, koska se tarjosi korkean tarkkuuden ja oli suhteellisen nopea käyttää. Sen kyky yhdistää useita päätöspuita mahdollisti monimutkaisten riippuvuuksien tunnistamisen ilman merkittävää ylikoulutusta.

MLP voi olla hyvä vaihtoehto, jos on aikaa ja resursseja optimoida mallia huolellisesti. Se osoitti potentiaalia erityisesti syvempien neuroverkkojen käytössä, mutta vaati huomattavasti enemmän hienosäätöä ja laskentatehoa.

Logistinen regressio ei soveltunut hyvin monimutkaisten verkkohyökkäysdatan luokitteluun, koska hyökkäykset eivät jakaudu helposti lineaarisiin ryhmiin. Tästä syystä malli teki paljon virheitä eikä tunnistanut hyökkäyksiä riittävän luotettavasti.

12 Yhteenveto

Tässä projektissa vertailtiin kolmea eri koneoppimismallia verkkohyökkäysten tunnistamiseen: Logistinen regressio, Random Forest ja MLP-neuroverkko. Mallien koulutus ja testaus suoritettiin CIC-IDS-2017-datasetin pohjalta, jossa oli sekä normaalia verkkoliikennettä että erilaisia hyökkäystyyppejä, kuten DDoS, PortScan, Brute Force ja SQL Injection.

Kokonaisuutena työssä onnistuttiin hyvin. Kolme eri koneoppimismallia koulutettiin onnistuneesti ja kaksi niistä tunnisti hyökkäyksiä hyvin. Random Forest osoittautui parhaaksi malliksi sillä sen hyökkäysten tunnistustarkkuus oli 99,87 % ja F1 score todella hyvällä tasolla. Se suoriutui erinomaisesti kaikista hyökkäystyypeistä ja oli lisäksi nopea kouluttaa.

MLP-neuroverkko saavutti hyvän tarkkuuden 99,48 %, mutta sillä oli vaikeuksia tunnistaa tiettyjä harvinaisempia hyökkäystyyppejä. Lisäksi sen koulutus kesti pidempään kuin muilla malleilla.

Logistisen regression tarkkuus tunnistaa hyökkäyksiä jäi muista selvästi. Se ei soveltunut monimuotoisen verkkohyökkäysdatan luokitteluun, ja sen suorituskyky oli erityisen heikko harvinaisempien hyökkäysten tunnistamisessa.

Benign (normaali liikenne) ja yleiset hyökkäykset kuten DDoS, DoS Hulk, PortScan tunnistettiin lähes täydellisesti kaikilla malleilla.

Harvinaiset hyökkäykset kuten SQL Injection, Brute Force, XSS olivat vaikeimpia tunnistaa. Näiden osalta Random Forest suoriutui parhaiten.

Harvinaisten hyökkäysten tunnistamisen haasteet johtuivat osittain testidatan epätasaisesta jakautumisesta, sillä tiettyjä hyökkäystyyppejä esiintyi vain muutamia kertoja koko datasetissä. Tämä vaikeutti mallien oppimista ja edellytti datan ja mallien hienosäätöä, mikä onnistui suhteellisen nopeasti.

Nykyinen tutkimus keskittyi offline-datasettiin, mutta seuraava askel voisi olla mallien soveltaminen reaaliaikaiseen verkkoliikenteeseen ilman viivettä. Malleja voisi myös yhdistää hybridimalliksi esim. Random Forest ja MLP, jolloin päästään hyödyntämään sekä päätöspuiden tarkkuutta että neuroverkkojen syväoppimiskykyä. TensorFlow-pohjaiset syväoppimismallit voisivat parantaa harvinaisten hyökkäysten tunnistusta, mutta niiden tehokas käyttöönotto vaatii enemmän resursseja ja optimointia.

Lähteet

Ahmad, Zeeshan. Khan, Adnan Shahid. Shiang, Cheah Wai. Abdullah, Johari. Ahmad, Farhan. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. 2020. Verkkoaineisto. https://onlinelibrary.wiley.com/doi/10.1002/ett.4150 Luettu 1.4.2025

GeeksforGeeks. Best Python libraries for Machine Learning. 2024. Verkkoaineisto. Luettu 31.3.2025

Cisco. What Is a Cyberattack? 2025. Verkkoaineisto. https://www.cisco.com/c/en/us/products/security/common-cyberattacks.html Luettu 30.3.2025

Fortinet. Cyber Threats. 2025. Verkkoaineisto. https://www.fortinet.com/topics/cyber-threats Luettu 29.3.2025

Intrusion detection evaluation dataset (CIC-IDS2017). Canadian Institute for Cybersecurity. 2017. Verkkoaineisto. https://www.unb.ca/cic/datasets/ids-2017.html Ladattu 29.3.2025

Keeper. How AI Is Making Phishing Attacks More Dangerous. 2024. Verkkoaineisto. < https://www.keepersecurity.com/blog/2024/09/13/how-ai-is-making-phishing-attacks-more-dangerous/?utm_source=chatgpt.com> Luettu 30.3.2025

Khraisat, Ansam. Gondal, Iqbal. Vamplew, Peter. Kamruzzaman, Joarder. Survey of intrusion detection systems: techniques, datasets and challenges. 2019. Verkkoaineisto. https://link.springer.com/article/10.1186/s42400-019-0038-7#Sec1 Luettu 1.4.2025

Marchal, Samuel. Bartosz, Nawrotek. WithSecure. Tekoälypohjaiset kyberturvallisuusratkaisut. Traficomin tutkimuksia ja selvityksiä 07/2024. Verkkoaineisto. https://www.kyberturvallisuuskeskus.fi/sites/default/files/media/file/Teko%C3%A4lypohjaiset%20kyberturvallisuusratkaisut_FI.pdf . Luettu 28.3.2025

Nmap. 2025. Verkkoaineisto. https://nmap.org/ Luettu 29.3.2025

Python. 2025. Verkkoaineisto. https://www.python.org/ Luettu 31.3.2025

OpenCV. 2025. Verkkoaineisto. https://opencv.org/ Luettu 31.3.2025

Rapid7. Network Traffic Analysis. 2025. Verkkoaineisto. https://www.rapid7.com/fundamentals/network-traffic-analysis/ Luettu. 28.3.2025

Reuters. Cyberattacks cost British businesses $55 billion in past five years, broker says. 2024. Verkkoaineisto. https://www.reuters.com/technology/cybersecurity/cyberattacks-cost-british-businesses-55-billion-past-five-years-broker-says-2024-11-25/?utm_source=chatgpt.com Luettu 31.3.2025

Scapy. 2025. Verkkoaineisto. https://scapy.net/ Luettu 29.3.2025

Schule, Mike. Comparing Manual vs. AI-Based Network Design: Efficiency in the Spotlight. 2025. Verkkoaineisto. <https://orhanergun.net/comparing-manual-vs-ai-based-network-design-efficiency-in-the-spotlight> Luettu 31.3.2025

State of IoT. 2024. Verkkoaineisto. https://iot-analytics.com/number-connected-iot-devices/ Luettu 26.3.2025

Tecnobits. Mitä ovat Legacy-järjestelmät ja miksi on yrityksiä, jotka eivät modernisoi teknologiaansa? 2025. Verkkoaineisto. https://tecnobits.com/fi/Mit%C3%A4-ovat-vanhat-j%C3%A4rjestelm%C3%A4t-ja-miksi-on-yrityksi%C3%A4--jotka-eiv%C3%A4t-modernisoi-teknologiaansa%3F/?utm_source=chatgpt.com Luettu 31.3.2025

Tcpdump. 2025. Verkkoaineisto. https://www.tcpdump.org/ Luettu 29.3.2025

Vähäkainu, Petri. Lehto, Martti. Neittaanmäki, Pekka. 2018. Tekoäly ja kyberturvallisuus. Informaatioteknologian tiedekunnan julkaisuja No. 60/2018. Jyväskylän yliopisto. Verkkoaineisto.< https://jyx.jyu.fi/jyx/Record/jyx_123456789_89189> Luettu 28.3.2025

Wireshark. 2025. Verkkoaineisto. https://www.wireshark.org/ Luettu 29.3.2025